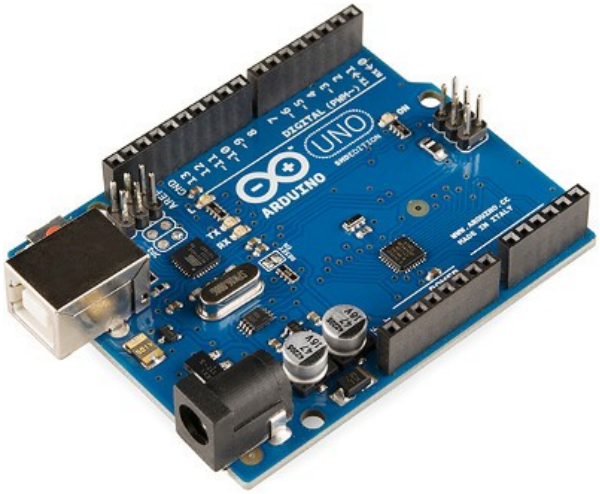


Scratch for Arduino
Lesson 2 – Dimmer Switch

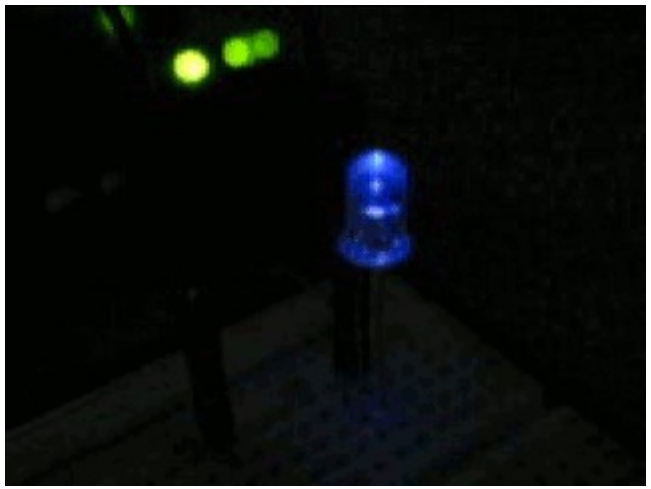
Variables, Loops, and Light Shows*

Light Shows may be omitted due to time constraints.

A POSTERIORI
Play · Experience · Learn

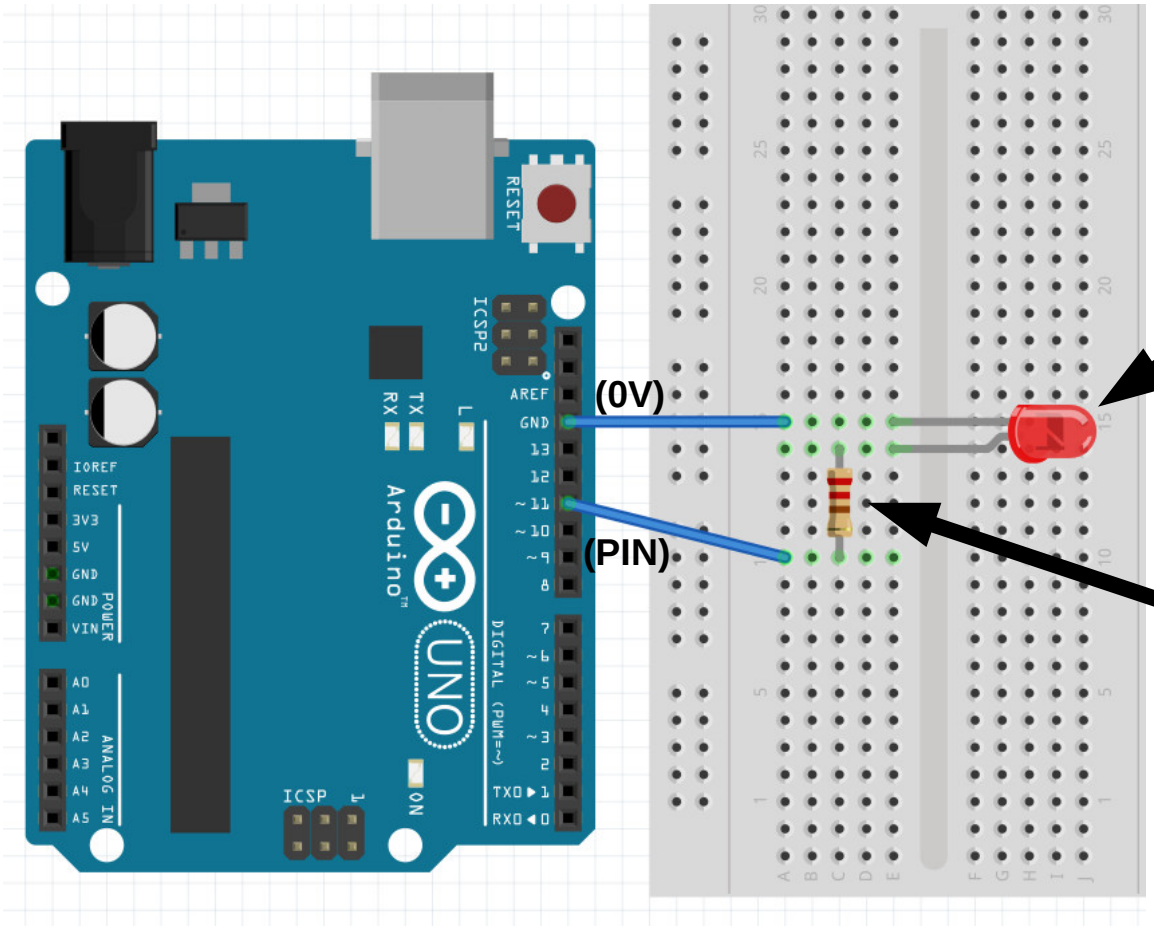
Light Control

- Today we will improve the Basic LED project by introducing dimmer effect



Slides available at: http://a9i.sg/huayi

Review LED Circuit



When PIN is set to LOW (0V), there's no voltage differential across circuit, so no current.

When PIN is set to HIGH (5V), voltage differential causes current to flow and light LED.

Slides available at: http://a9i.sg/huayi

Controlling Brightness

So, how can we control Brightness?

- Change Resistor
  (not really programmable, but let's **revisit in the next lesson!**)

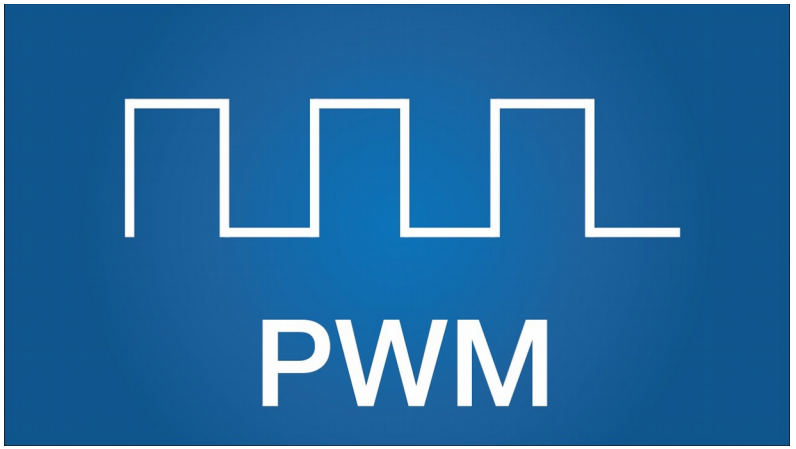
- Change Power
  But, all of our Output pins are **Digital (On/Off)**

We need PWM!!!

What is PWM?

Pulse Width Modulation (PWM)

- An efficient method to vary and control power
- Used in various electrical systems
  - Lights
  - Motors
  - Comms & others



Change Power

- We can On/Off or HIGH/LOW or 5V/0V

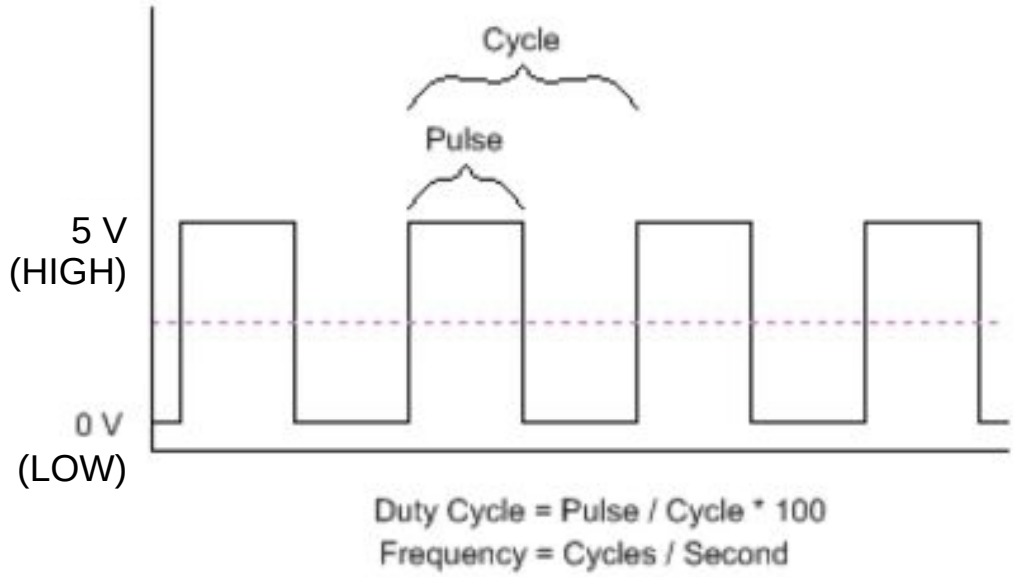
  But, what if we wanted **%50 power**?

- Can’t set digital output to 2.5V….

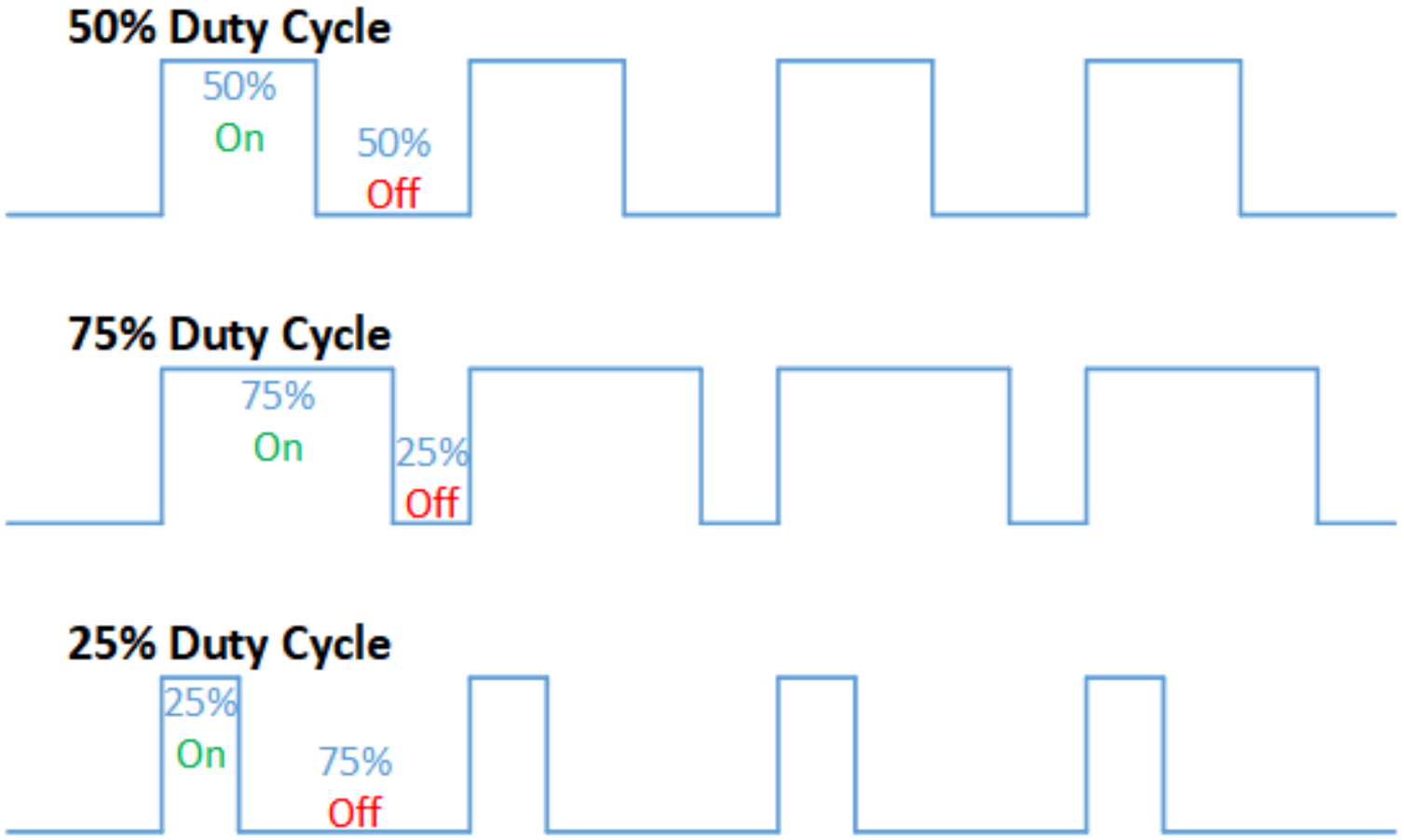
  - So, add ***Time*** to the equation!

%50 Duty Cycle

- Pick some Interval or Period (1 ms, or 1000 Hz)
- Set signal to HIGH ½ of the period (0.5 ms)
- Set signal to LOW rest of the period (0.5 ms)
- Run signal over and over...

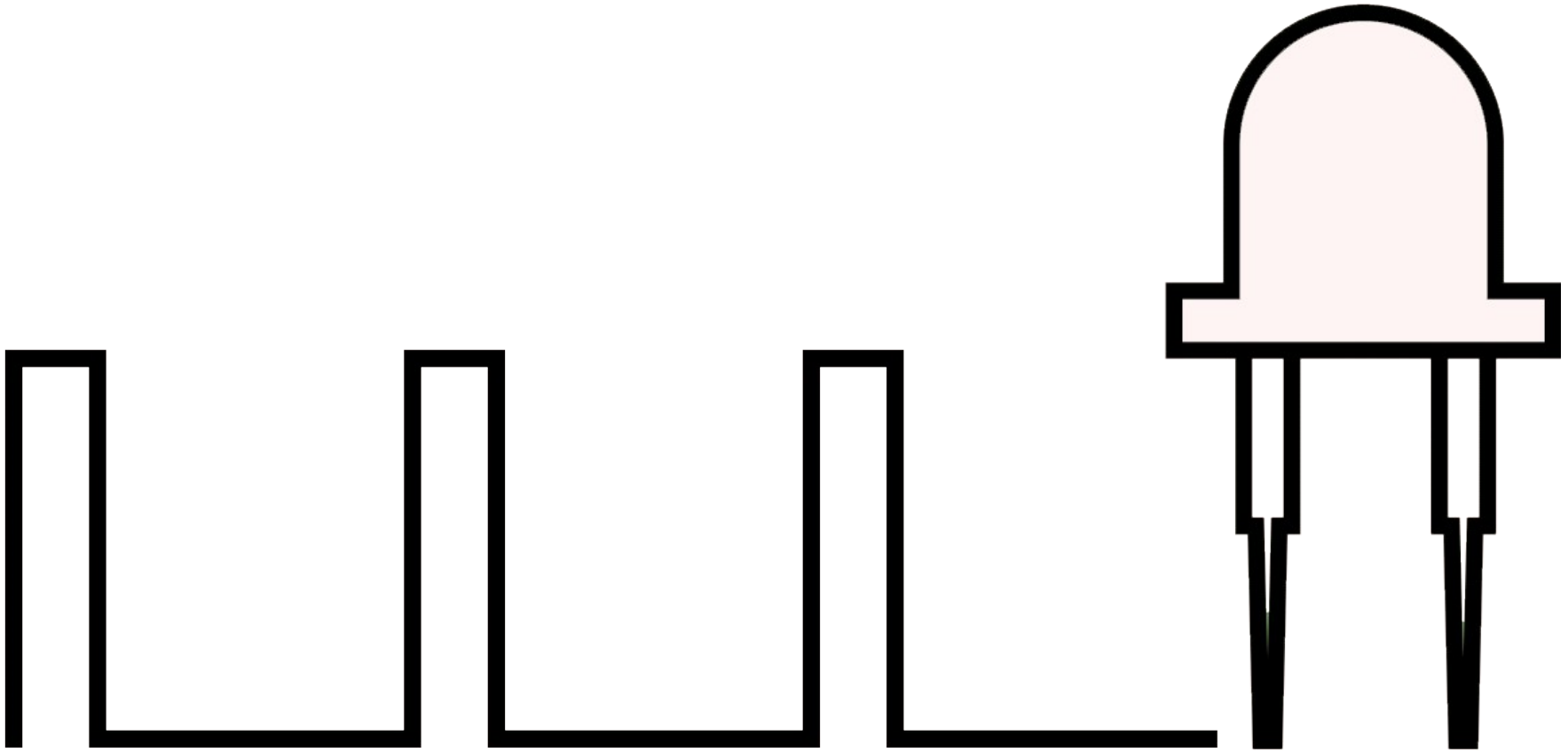


Duty Cycle



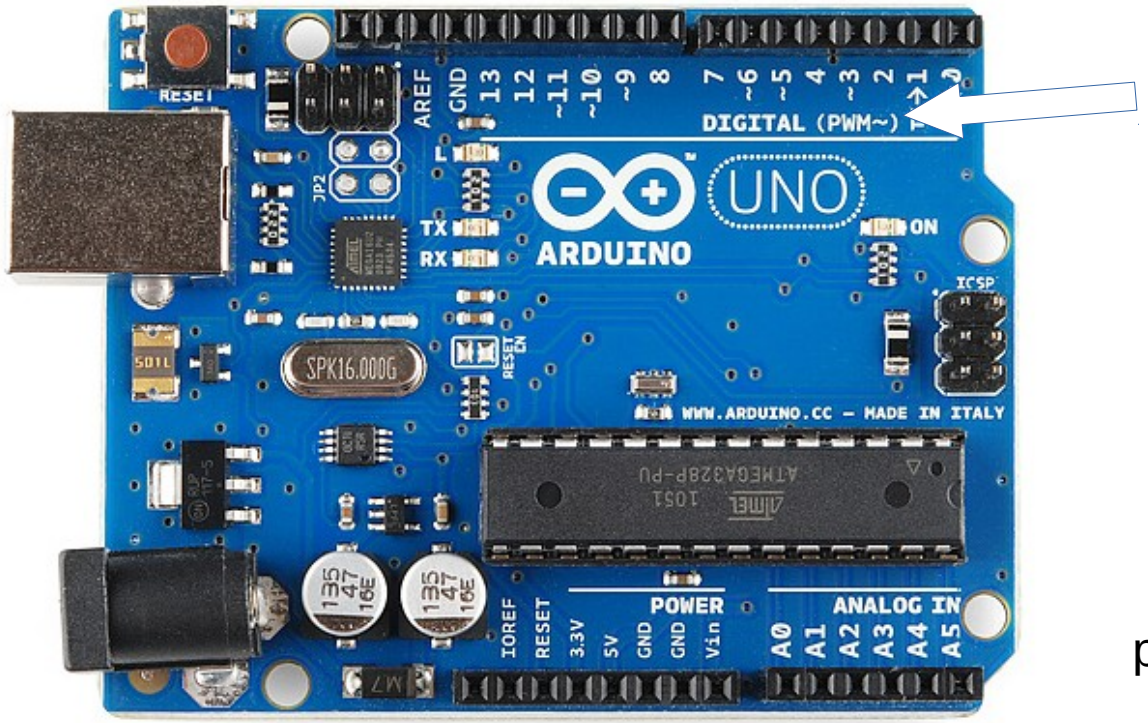
Slides available at: http://a9i.sg/huayi

Duty Cycle



PWM Pins

- Today we'll work with Pulse Width Modulation (PWM)
- Take note which pins support PWM



The board usually indicates which pins have PWM built-in support

In this case Pins 3,5,6,9,10,11 (see '~')

All output pins are **digital** (LOW/HIGH), so we can only vary power using something like the PWM method

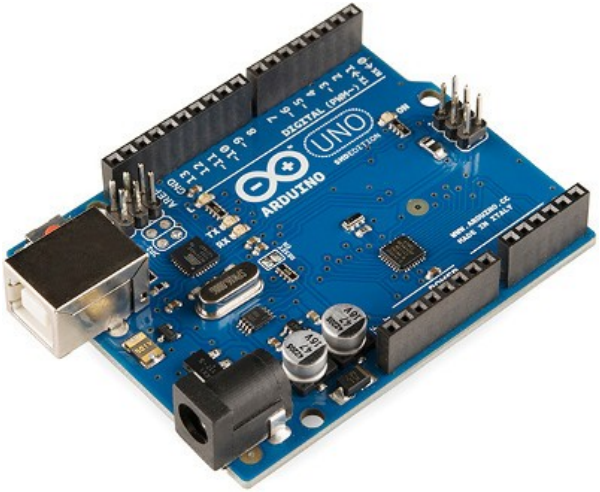
Let’s Get Hands-On

(The fun part…?)

A POSTERIORI

Play · Experience · Learn

Physical Connection



Some of the LED on the Arduino should light up after connection

This end to computer

This end to Arduino

Slides available at: http://a9i.sg/huayi

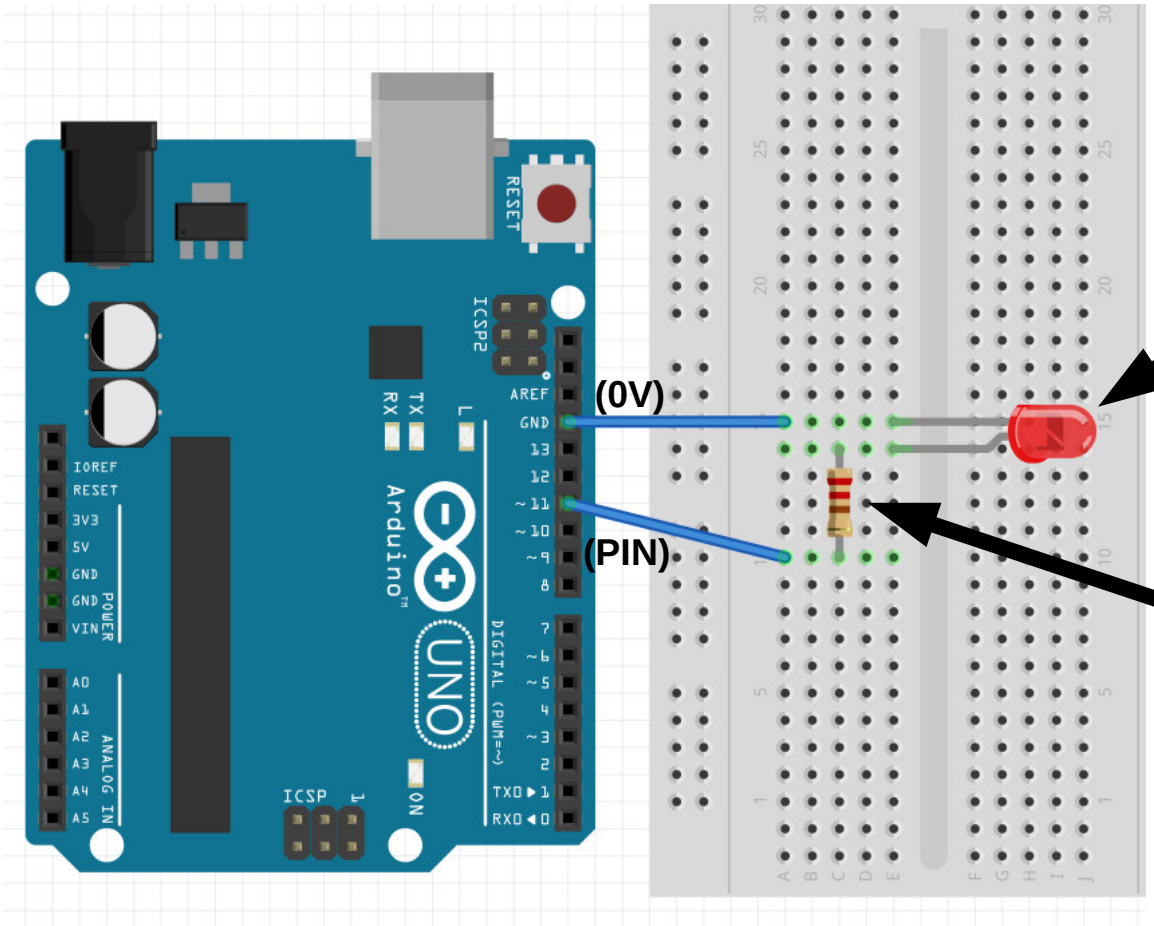
Exercise 2a

Control LED Brightness

A POSTERIORI

Play · Experience · Learn

Review LED Circuit



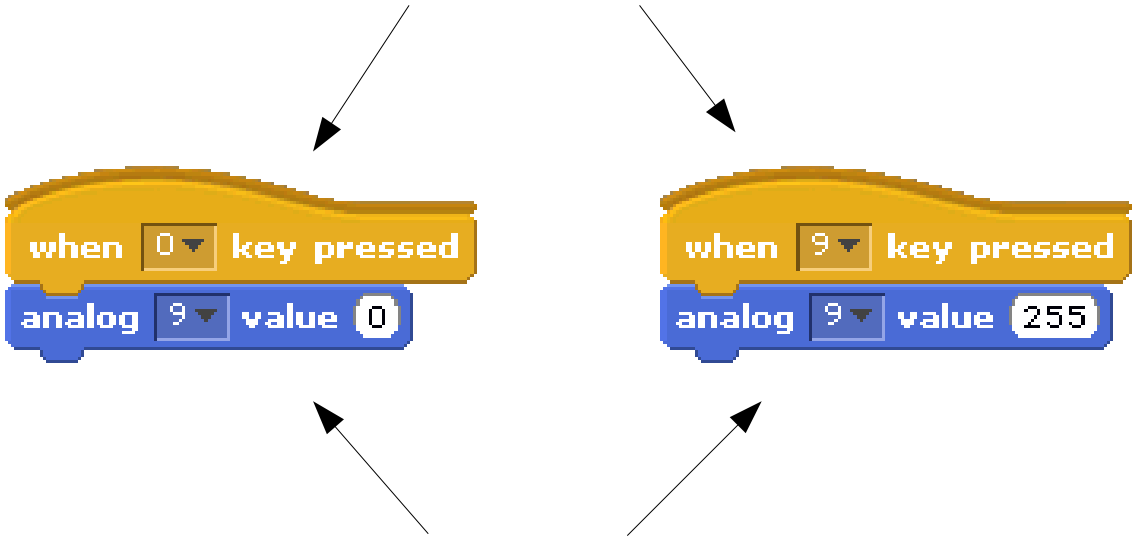
When PIN is set to LOW (0V), there's no voltage differential across circuit, so no current.

When PIN is set to HIGH (5V), voltage differential causes current to flow and light LED.

Slides available at: http://a9i.sg/huayi

Set PWM on LED Pin

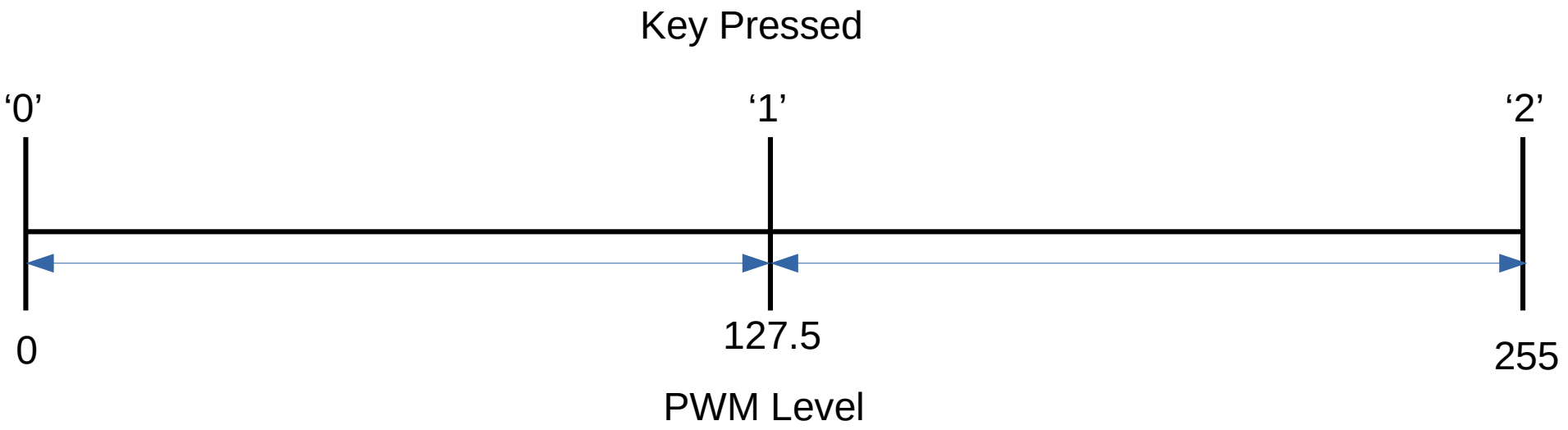
Let's use Keys 0-9 to set Brightness Levels.
Find the **When-Key-Pressed** block under **Control**.
For starters, let's deal with our limits – 0 (off) and 9 (full power)



Find the **analog** block under **motion.**
Set the Pin to the one which you connected your LED Long(+) leg.
(*you **have** to connect pin 5,6, or 9 for analog output*)

Test it out!

Challenges



For 3 levels – **off, medium, high** –
we would use this key mapping...

Set PWM on LED Pin

Let’s add 1 extra level for medium power.

0 for ‘0’

LED
Should be off

127.5 for ‘5’

LED
Should be lit
but dim

255 for 9

LED
Should be lit
brightly

Test it out!

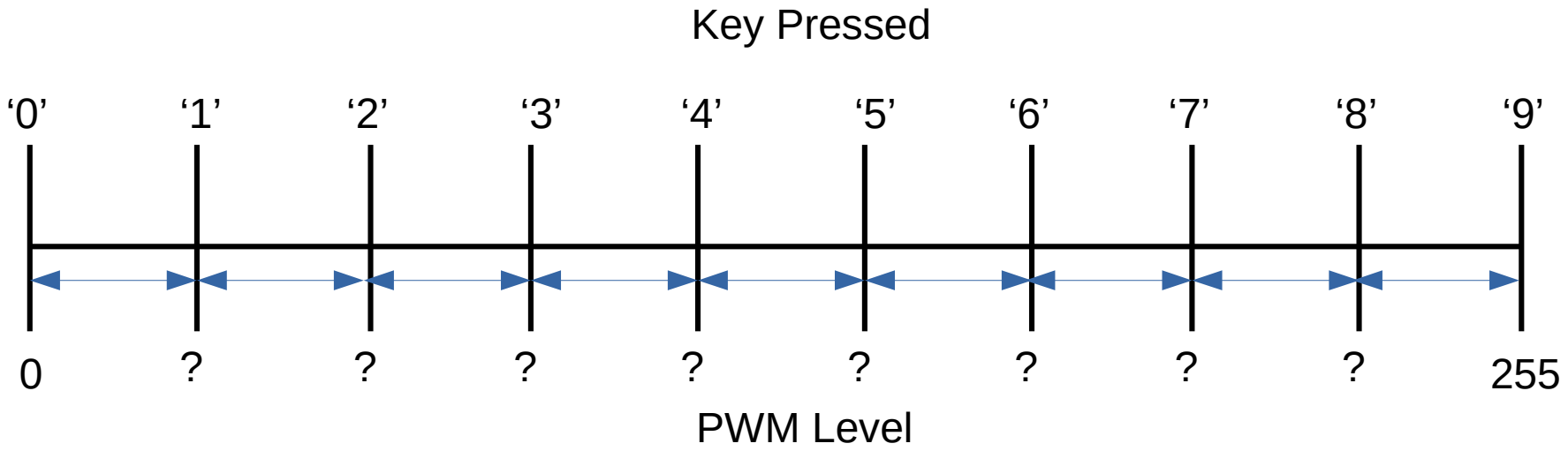
Exercise 2a

Control LED Brightness

A POSTERIORI

Play · Experience · Learn

Challenges



What would be the appropriate PWM levels for each Key Press ‘0’ - ‘9’?
(10 power levels)

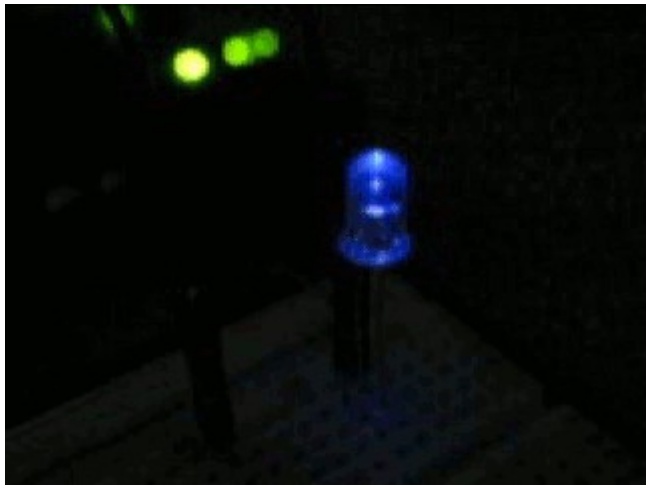
Challenges

- In your Student Handout
  - Note down the **PWM levels** you used for all 10 keys
  - Note down **math formula** to generate appropriate PWM levels for each user input (0-9)
  - Generalize for **N number of inputs**

- Add code to control LED brightness levels using **all ten digit (0-9) keys**

Light Control

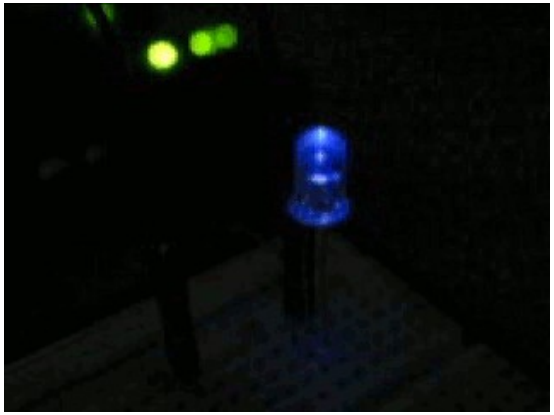
- Back to our dimmer effect...



- Can you use your program to create this effect?

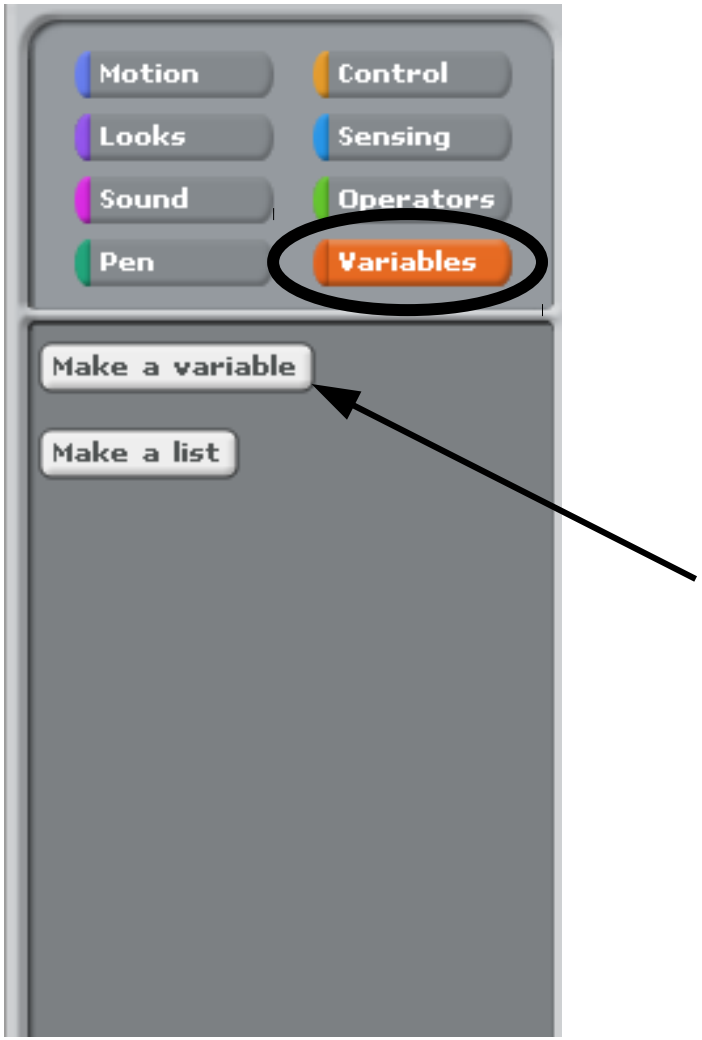
Light Control

- Sort of...
  If we want a smooth dimmer effect we need to run LED through all/many PWM levels 0-255



- But we don’t have enough keys or patience to press all of them, so let’s try something new!

Variables



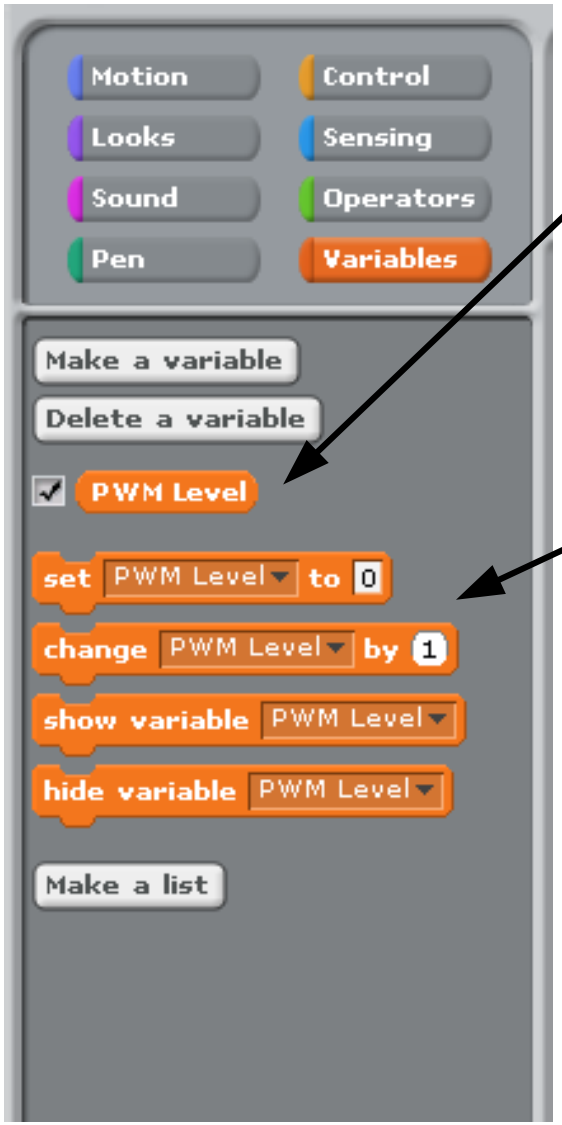
- First, Let's create a new Variable.
  A variable is a named piece of memory that stores information like numbers or strings.
  Its value can change, hence it is variable…
- Click on "Make a Variable"
  Under **Variables** menu

Variables

Pick a meaningful Name

Doesn’t matter much for today,
but keep it “For all sprites”

Variables



Now we have a new variable!

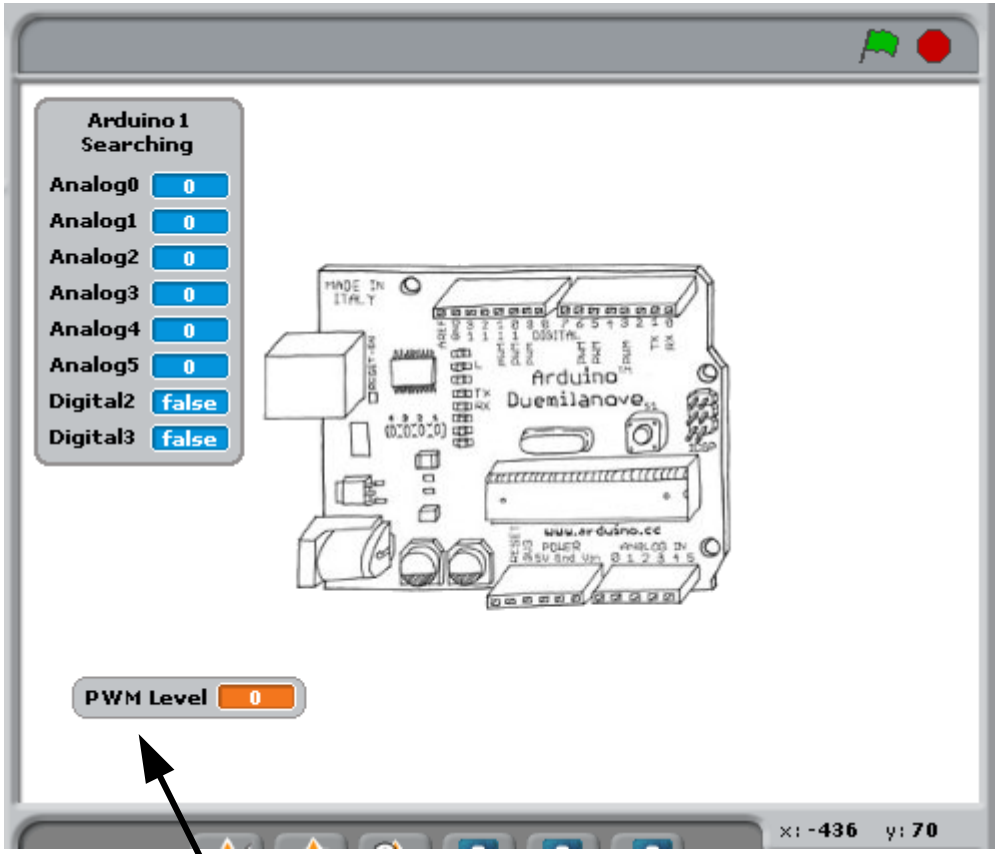
These are helper function blocks associated with variables:

“set Var to 0” → “Var = 0”

“change Var by 1” → “Var = Var +1”

Let’s use them now...

Variables



Let's play with this variable...

We can change the value in a simple program, and see the Display change

To test, keep changing this value and clicking **space** key to change the variable.

Light Control - Variable

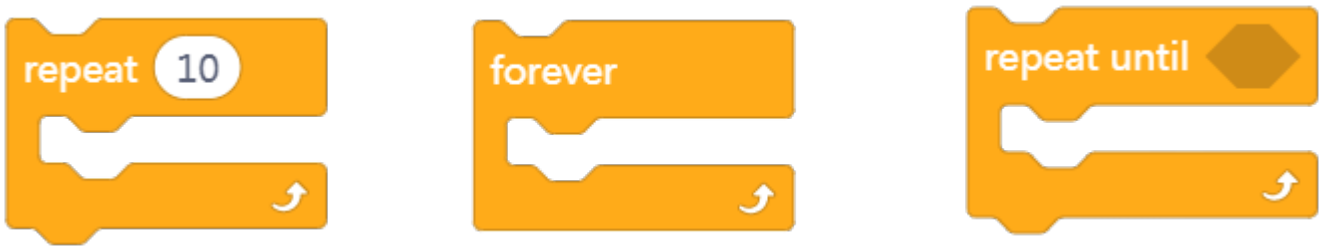
We want to use the value in our program to change the PWM output of our LED pin

Any ideas how?

Light Control - Loops

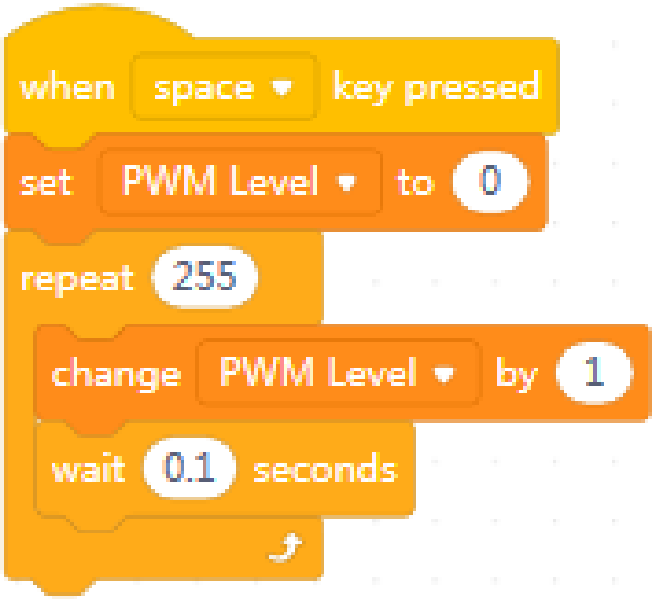
What about the Dimmer Effect?

We Use Loops



Light Control – Loops & Variables

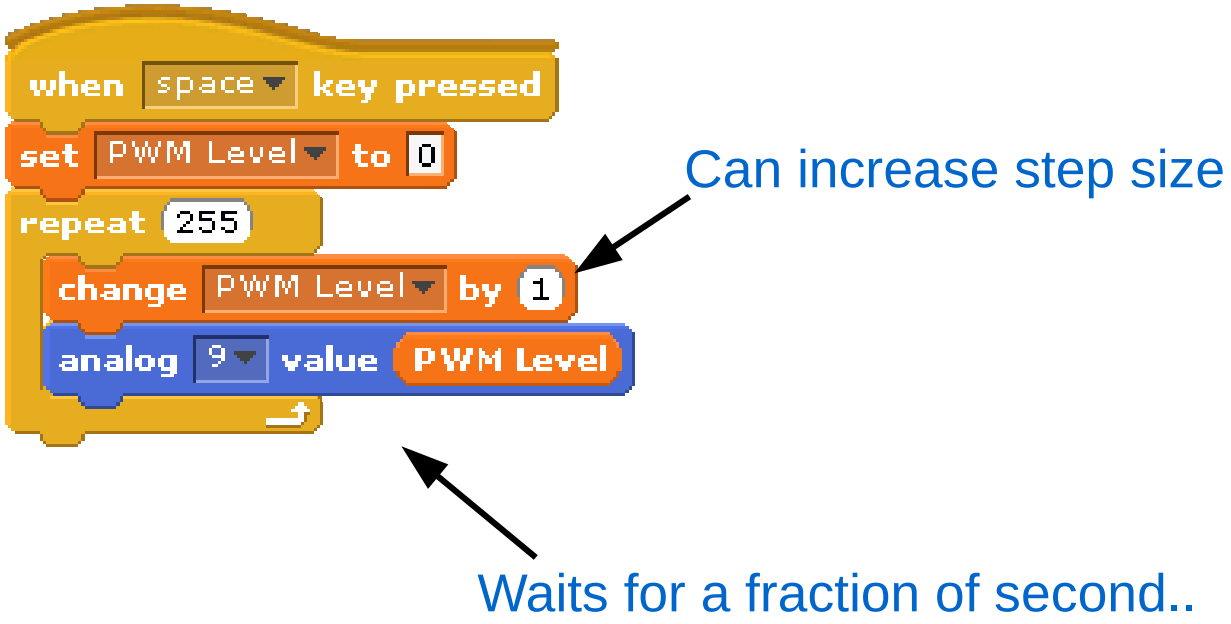
For instance, let's make a simple counter:



See the variable value change:

Dimmer Effect

- Instead of **wait** block, set the LED pin to the increasing PWM levels



Exercise 2b

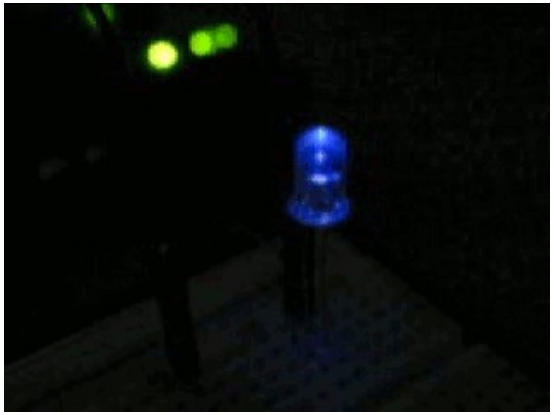
Create a Smooth Dimmer Effect

A POSTERIORI

Play · Experience · Learn

Challenges

- After trying various steps, note the **Change By [value]** that gave you the best dimmer effect in your Student Handout
- Extend the program to make dimmer effect run from low to high **and back to low again**
- Make the program run continuously like this:



Extra Challenges

- Convert your variable to a **Slider** and use it as a graphical **Variable** Dimmer Switch
- Create a Graphical **Dashboard** to control Lights (on/off buttons, slider dimmers, blink buttons)
- Use a physical button to act as a **Toggle** Dimmer Switch (on/off)
- Use multiple LEDs to create a **Light Show** with blinking, dimming, and any other effects you can muster

Copyright

- Created by A Posteriori LLP
- Visit http://aposteriori.com.sg/ for more tips and tutorials
- This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.